

# A workflow for software development within computational epidemiology\*

Baki Cakici<sup>1,2</sup> and Magnus Boman<sup>1,3</sup>

{cakici, mab}@kth.se

## Abstract

A critical investigation into computational models developed for studying the spread of communicable disease is presented. The case in point is a spatially explicit micro-meso-macro model for the entire Swedish population built on registry data, thus far used for smallpox and for influenza-like illnesses. The lessons learned from a software development project of more than 100 person months are collected into a check list. The list is intended for use by computational epidemiologists and policy makers, and the workflow incorporating these two roles is described in detail.

## 1 Introduction

### 1.1 Computational epidemiology

In 1916, Ross noted that mathematical studies of epidemics were few in number in spite of the fact that “vast masses of statistics have long been awaiting proper examination” (page 205, [1]). In the 90 years which followed, the studies made were analytic, and the micro-level data available were largely left waiting, to leave room for systems of differential equations built on homogeneous mixing. This is remarkable not least because the modeling problem remains the same throughout history: “One (or more) infected person is introduced into a community of individuals, more or less susceptible to the disease in question. The disease spreads from the affected to the unaffected by contact infection. Each infected person runs through the course of his sickness, and finally is removed from the number of those who are sick, by recovery or by death. The chances of recovery or death vary from day to day during the course of his illness. The chances that the affected may convey infection to the unaffected are likewise dependent upon the stage of the sickness.” (page 700, [2]). Heterogeneity is present already in this classic description, in several places; susceptibility, morbidity, and also contact patterns, if only implicitly. Only with the advent of

---

<sup>1</sup>Royal Institute of Technology (KTH/ICT/SCS), SE-16440 Kista, Sweden

<sup>2</sup>Swedish Institute for Communicable Disease Control (SMI), SE-17182 Solna, Sweden

<sup>3</sup>Swedish Institute of Computer Science (SICS), SE-16429 Kista, Sweden

\*Please cite as: Baki Cakici, Magnus Boman, “A workflow for software development within computational epidemiology”, *Journal of Computational Science*, Available online 6 June 2011, ISSN 1877-7503, doi:10.1016/j.jocs.2011.05.004

powerful personal computers, were micro-level data given a role in the modeling of epidemics. Executable simulation models in which each individual could be modeled as an active object with its own attributes [3], often referred to as an agent, began to appear [4, 5, 6]. A new area within computer science, computational epidemiology, has recently become established as the scientific study of all things epidemiological except the medical aspects. This area is turning into computational science (see, e.g., [7]), following the example of computational biology, computational neurology, computational medicine, and several other new areas focusing on building computationally efficient executable models. This development also includes the social sciences, as in computational sociology [8].

## 1.2 Model description

The model on which the analysis below is based has been continuously developed since 2002 by a cross-disciplinary group of researchers from the fields of medicine, statistics, mathematics, sociology and computer science. Since 2004, a team of developers have implemented various versions of a software tool, representing the computational part of the model, recently made available as open source software and licensed under GNU General Public License Version 3 [9]. In parallel with the implementation, the requirements on the model have changed many times. It began as a model for predicting the effects of a possible smallpox outbreak in Sweden [10], which was later transformed into a model for studying pandemic influenza, and is now a model that could be used for many different kinds of communicable disease studies (excluding vector-borne diseases, i.e., diseases with animal reservoirs). The model is a detailed representation of real situations, sometimes referred to as a tactical model, as opposed to simpler strategic models [11]. For instance, the model was recently used to study a fictitious scenario of H4N6: a new influenza virus strain that was assumed to be deadly, highly contagious, and introduced into a completely susceptible population. In all, the development project has included more than 100 person months of implementation work, and consists of more than 5000 lines of C++ code.

The parameters used to represent individuals in the model are age, sex and current status (alive or deceased). Each individual is also assigned a home, a workplace, and a department within that workplace. The movement of individuals outside of home and workplace are represented using travel status (home or in another location), emergency room visits, and hospitalizations.

Infections caused by social contact outside of work or home are classified as context infections. When the context infection process is active, there is a probability that an infectious individual will infect those that live within a fixed radius. Context contact radius defines the size of neighborhoods, mirroring the interaction of every individual with others, based on geographical proximity and the social network.

The disease affects every individual through three parameters: infectiousness, death risk, and place preference. The infectiousness parameter influences the probability that the infected individual will infect others in the same home, workplace, or neighborhood. The death risk depends on the disease level and is expressed as a probability. Place preference is the probability distribution used when deciding where the individuals will spend their day (workplace, home, pri-

mary care, or hospital). These parameters are defined for five levels of severity: asymptomatic, mild, intermediate, severe, and critical. In addition, there are four disease profiles: asymptomatic, mild, typical, and atypical.

The model description is combined with Swedish data on workplaces, households, and individuals. Workplaces include companies, schools, healthcare, and other state institutions. For each workplace, the data indicate the total number of workers, geographical coordinates, and workplace type. The current version of the simulation platform uses data from the Swedish Total Population Register, the Swedish Employment Register, and the Geographic Database of Sweden (cf. [12]).

Because the model was developed with the purpose of being run with data for the country of Sweden, it has been used solely for studying outbreaks in that country. Sweden has relatively many infection clinics and good international reputation for detailed clinical reports of communicable disease. Thus, in some areas of disease control, Sweden works well as a role model. Other countries face special local problems, however, and results have sought to be generalizable, for example contributing to the complicated model of EU care-seeking behavior. Generally speaking, the project goals have included to sensitize policy makers to the scope of possible disruption due to a newly emergent disease event, and to identify a range of policy handles which can be used to respond to such an episode.

A sample case description illustrates how an experiment would be described using the executable model. The sample case simulates the effects of pandemic influenza in Sweden, without any interventions, for 300 days. The simulation is initiated with 50 infected individuals, randomly selected from the entire population. Since the data set is registry data for the entire country, any random selection procedure is uniform, i.e., an individual has a 50 in nine million chance of being initially infected. This does not mirror realistic spread, which would more typically be an airplane or a boat arriving to Sweden with one or more infected individuals on board, but in the sample case it at least provides an opportunity to discuss the complex matter of how epidemics start. The maximum size for an office is set to 16 individuals and all workplaces with more than 16 employees are split into departments, each containing 16 or fewer members. This value is not arbitrary, but corresponds to the average size of a Swedish workplace. Context contacts – the parameter representing the average number of contacts outside the home or the workplace – is set to 15. Even if that number was recommended by the sociologists in the project, it is somewhat arbitrary, and is therefore subjected to sensitivity analyses in our sample case. Naturally, such analyses would be extensive in a real policy case; here the reason for their inclusion is chiefly pedagogical.

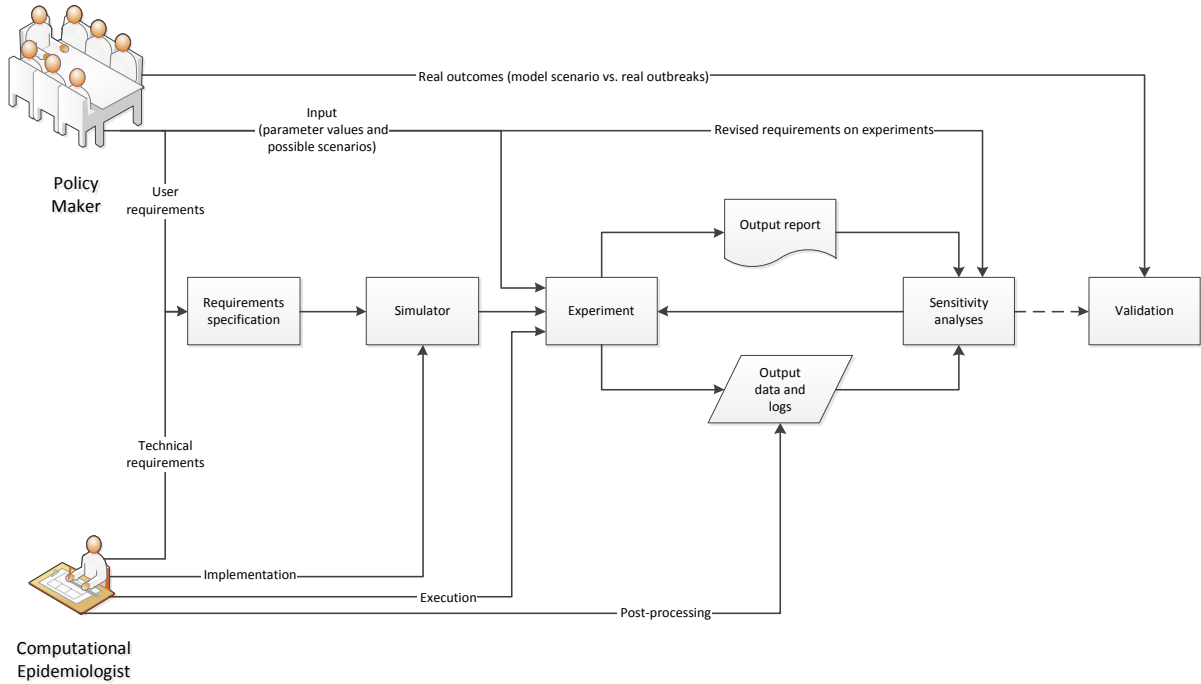
### 1.3 Disposition

A report on lessons learned from the software development project constitutes the bulk of the analysis below. It starts with a description of the workflow in a computational epidemiology project, and observations on the micro-meso-macro link follow. More detailed descriptions of what it actually means to manage and run a simulator are then provided, before discussing the scientific merits and challenges of this kind of research, and the concluding check list is presented.

## 2 Workflow

### 2.1 Model development

The process of developing a model for outbreaks today often includes the development of a simulator, allowing for scenario execution and relatively swift sensitivity analyses. The simulator does not capture the entire model, but only those parts that are subject to uncertainty or those that involve stochastic parameters. The instigator is typically a policy maker (PM), knowledgeable in public health issues, and seeking to evaluate various scenarios. The PM may well have medical training, or even be an epidemiologist. The implementer of the simulator is a computational epidemiologist (CE): a modeler knowledgeable in computer science, and the social sciences, typically without much medical training. Naturally, both PM and CE could denote a team instead of a single person. A schematic workflow for developing and using a simulator, depicting the roles of both PM and CE, is presented in Figure 1.



**Figure 1:** The schematic workflow of developing and running an executable model, incorporating policy makers and computational epidemiologists.

As in all development projects, work begins with a requirements specification, to which the PM contributes user requirements and the CE contributes technical expertise. From this specification, the simulator is built. It consists of a software package with two parts: a simulation engine and a world description. The latter is not the complete description of the world under study, but covers only those parts that have a bearing on the executable model. This modeling work is carried out by the CE, with considerable assistance from medical professionals. The CE implements the simulator in accordance with the specification and medical expertise. The CE will also seek to verify the accuracy of the simulator (e.g., through extensive testing, or even logical proof). The CE works in two distinct sequential steps that cannot be combined: design and implementation. Software engineers are taught not to modify their design during the implementation stage to “improve” the model, no matter how tempting this might be. If design decisions leak into the implementation stage, the software project quickly becomes impossible to maintain. What software design means in the area of computational epidemiology is the craft of knowing which parameters to vary, being aware of their mutual dependence, and how to openly declare all simplifying assumptions.

Once the simulator is complete it is given a version number, and one may proceed to experiments. For an experiment to be meaningful, the PM must envisage scenarios. The PM must also provide values for some input parameters. Each parameter in the model is important, and even slight changes to an input value might have a drastic effect on the output. The kind of model considered here is a complex system: a system which cannot be understood through understanding its parts. Before the CE can run the system, the world description must be populated with data, which typically need a significant amount of post-processing to allow for smooth use in the simulator. In addition, one must then attempt to ascertain that the resulting data set is accurate and noise-free. The data set in the here described model was sensitive with respect to personal integrity, as it consisted of registry data on the entire Swedish population of approximately nine million individuals. This sensitivity rendered many kinds of replication experiments impossible.

Once the system runs, it will produce a vast amount of output, so experiments must be set up carefully to avoid information overload. The so-called induction trap – the lure of running too many experiments for each scenario because it is easy to produce more output, and then jumping to inductive conclusions too swiftly [13] – must also be avoided. The output and logs of a set of runs typically do not lend themselves to straightforward reading, but require post-processing. In practice, this means turning huge text files into calculable spreadsheets, and further into graphs and diagrams. Those outputs can then be presented back to the PM, who can then call for more experiments, sensitivity analyses, or even a revision of the requirements specification. The CE in this process makes certain design choices, e.g., which output data to present and how. It is important that this process is iterative and that the PM is given the option of making informed choices, by having at least some grasp of what is realistic to do, given the constraints of computational complexity. The CE must provide technical specifications on further experiments, and the technical competence used also comes with a responsibility to inform: the PM must know what options there are, and why and how certain results were omitted or deemed irrelevant. Because the PM is typically the one responsible for acting

upon results obtained, a chain of trust to the CE must be upheld. Likewise, the CE should react if the PM, for example, calls only for certain experiments to be run, or if the selection is made so as to confirm a preconceived truth, in a pseudo-scientific fashion [14].

In principle, the output of the executable model can finally be validated by comparing its predictions to real outcomes of actual policy interventions for the population modeled, given that the input parameters adequately model the real population prior to those interventions. Naturally, some scenarios could be considered extreme (e.g., the introduction of an entirely new influenza virus to a population without native immunity) and are simulated precisely because they cannot be studied in the real world. In such scenarios, validation can, at best, pertain only to parts of the model. More importantly, simulations of outbreaks are difficult to validate because the simulated event is rare. Catastrophic events are characterized by low probability and disastrous consequences (see, e.g., [15]), and yet the input data are collected from the normal state of the population in non-outbreak situations. Using this input, the simulator is expected to produce one possible yet highly unlikely scenario to provide researchers and policy makers with more opportunities to observe and learn about the unlikely event.

Since computational epidemiology is problem-oriented and constitutes applied science, models are often pragmatic in the sense that they are adapted to their use as policy-supporting tools. Any provisos made have to be grounded in the culture of the decision making entity, such as a government or a pharmaceutical company, making alignment studies, in which models are docked for replication studies [16] difficult.

## 2.2 The micro-meso-macro link

In microsimulation models of outbreaks, individuals are exposed to the disease and may infect other individuals that they come into contact with. The most primitive unit is the individual and the focus is on the activities of the individual, for the purposes of studying transmission. By contrast, macrosimulation focuses not on the individual, but on the whole society. All members (i.e., the whole, possibly stratified, population) share the same properties and move between different disease states such as susceptible, infected, and resistant.

Even if originally conceived as a pure microsimulation model, the executable model discussed here has macro-level parameters, e.g., workplace size. This parameter governs how many colleagues a working individual interacts with during a working day. To “interact with” here means that there is an opportunity for infection, given that either the individual or the colleague is ill. Even though micro data are available for each workplace – including the number of employees at each company – it is defensible not to use these data in full, since large workplaces have so many employees that it makes no sense to assume that the individual interacts with them all. In reality, the individual might not even see more than a fraction of the total number of colleagues on a given day. The workplace size is therefore set to a precise value, meant to capture an average number of colleagues, which is kept constant throughout a set of runs.

By definition, macro models do not represent local interaction. However, in any dynamic model utilizing micro data, including SIR-inspired individual-based models [17], local interaction will affect the output. If there appear discernible patterns in the output that are not explicitly stated by the model descrip-

tion at the outset, they are referred to as emergent patterns. In the described model, all output logs are mapped onto a real population. This means that every discernible pattern has an interpretation that can be understood in the epidemiological context, using terms such as “spread” and “giant component”, and also in the societal context, using terms like “number of infected” and “absenteeism”. Hence, patterns discernible at the macro level resulting from local interactions at the micro level are easily made understandable to the PM.

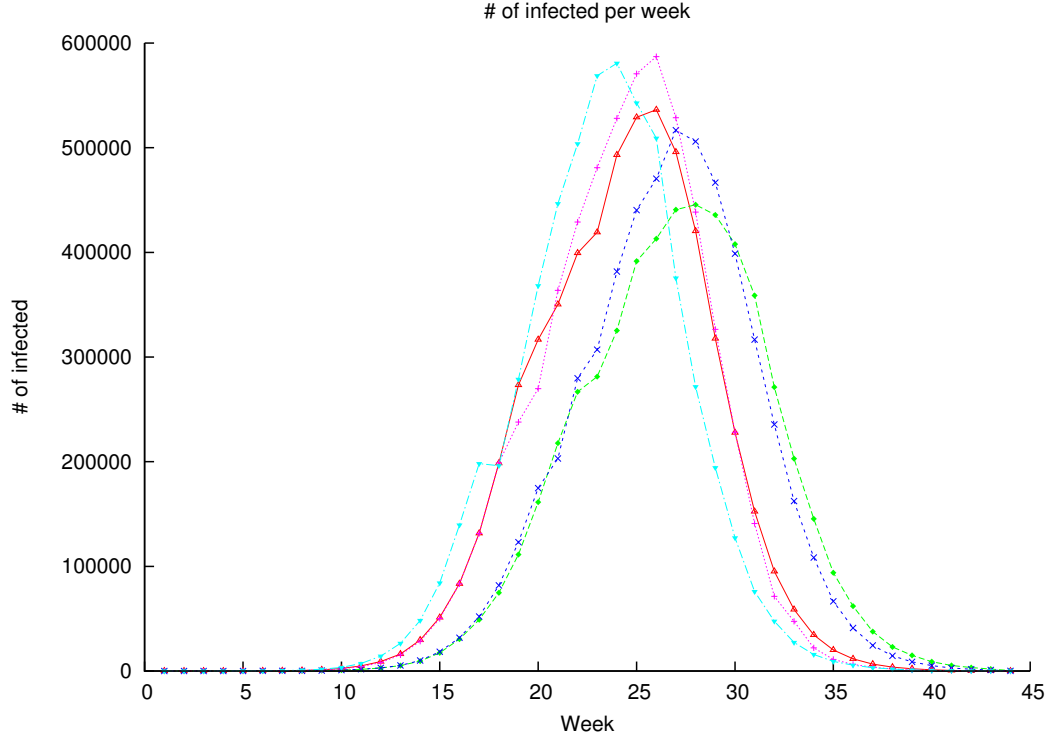
The meso layer [18] includes everything that is more general than the properties of single individuals but less general than the properties of the whole society. In the model at hand, this is most visible in neighborhoods, defined by the geographical proximity of different households. Adding the meso layer to an epidemiological model enables researchers to represent a crucial part of human interaction: social contacts outside the home or workplace. This includes encountering others while shopping, and social gatherings of neighbors.

Variables in the executable model represent properties of the real population, but many of them cannot be observed directly. Therefore, the argument goes, a suitable value for the executable must be determined by experimenting with the simulator. In the implementation phase, the CE strives to get a handle on the parameter space, i.e., the value space for all parameters that can be subject to variation. To illustrate this, a sample case is now considered.

To find a suitable value for the parameter *context contacts*, representing the average number of contacts outside the home or the workplace, the behavior of the simulated outbreak is observed using the total number of infected individuals per week for a large interval of context contact values. The interval is set to start from zero, where the model behavior is undefined, to where the parameter no longer has an observable impact, i.e., when it is high enough to exhaust the population regardless of all other parameters. Within the [8,20] interval, changing the *context contacts* parameter had, in this example, a significant effect on the behavior of the model. Repeating the same series of experiments with a smaller step size within the [8,20] interval, a smaller region of interest was obtained within the [14,16] interval. Finally, the analysis was repeated one last time for the [14,16] interval with a smaller step size. Figure 2 shows the number of infections per week for five runs where all parameters except *context contacts* were kept constant. Further simulations were run to observe the effects of variation due to random seeds when contacts was set to 15. Figure 3 shows the number of infections per week for three runs with different random seeds where all other parameters were kept constant. Other variables in the executable model that should be decided using a similar process include (but are not limited to): number of initially infected, office size, place choice based on disease level, place choice based on age, length of a work day, and the probability of receiving a symptomatic disease profile.

### 2.3 Stochasticity

An outbreak of pandemic influenza is a rare event. To trigger such an outbreak, either the simulations must be run repeatedly for a long period until an outbreak occurs, or the model must be configured in such a way that outbreaks will occur with higher frequency than in the real world. The former is not practical since it might take millions of runs before anything happens, and the latter comes with the risk of compromising the validity of output by introducing exogeneous



**Figure 2:** Number of infections per week for five runs where all parameters except context contacts were kept constant.

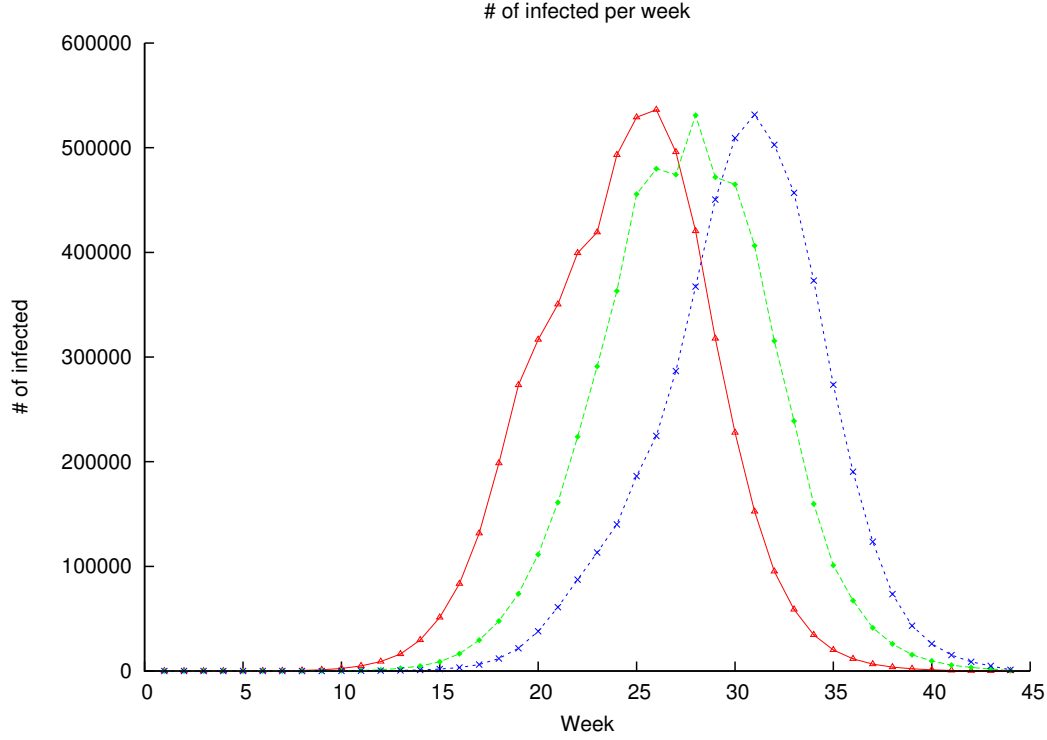
variables that change the effects of the simulated outbreak.

All random events in the model use a series of numbers that are generated at run-time using the initial seeds provided by the user. Therefore, the outcome of every “random” event in a simulation run depends only on the initial seeds. By using the same seeds, identical results can be obtained using different computers, operating systems, or compilers.

In the present model, one highly influential parameter is the number of initially infected. When 50 randomly selected individuals are infected, an outbreak is triggered in nearly every run. If only three individuals are selected instead, the outbreaks become much more rare. This is due to the heterogeneity of the population: individuals with more contacts are more likely to initiate outbreaks if infected, and it is more likely that a highly connected individual would be infected if 50 rather than three are infected initially.

It is often assumed in executable models that in a few generations, a simulation with three infected would reach the stage with 50 infected, and that the difference between them would be negligible. Certainly every simulation





**Figure 3:** Number of infections per week for three runs with different random seeds where all other parameters were kept constant. Each random seed is a vector of numbers generated by a pseudo-random number generator.

with three initially infected would reach a stage with 50 infected, given that an outbreak occurs during the run. Therefore simulations can be started from the stage where 50 individuals are infected since that is the minimum number at which the simulation platform produces outbreaks in the majority of runs. This assumption is far from ideal. The simplest observable effect is that no runs will have less than 50 infected. This is acceptable because the object of study is nation-wide outbreaks. However, the difference between the two approaches is not negligible because 50 randomly selected individuals will not have the same geographical distribution as 50 individuals whose infections originate from three individuals. The 50-from-three group will most likely have overlapping social networks because they were all infected by three individuals, as opposed to being randomly selected from a population of nine million. As the outbreak grows to one thousand or one hundred thousand infected, the difference may lose its significance, but quantifying that significance remains challenging for all

executable models that use heterogeneous populations. Hence, this is a good example of a simulation in which the CE makes an assumption about things beyond the PM’s control, or even grasp. Good software development requires that such assumptions be made explicit and communicated to the PM.

### 3 Conclusion

The lessons learned from the software development project described above can be summarized in the form of a check list. Even if the list is not exhaustive, developers of computational epidemiology models could check off the items on the list, as applicable to their project. The presented workflow and checklist do not include surveillance in computational epidemiology and instead focus on modeling and simulation. A more comprehensive workflow for computational epidemiology would have to incorporate computer-assisted infectious disease surveillance, often performed using complex software platforms tailored to the task [19, 20, 21, 22], and the interaction of its users with the actors already identified in the preceding sections.

Computational epidemiology is a new area, and many of the methods and theories employed have yet to benefit from thorough scientific investigation. Even if important steps towards amalgamating models and performing alignment experiments have been taken (see, e.g., [23]), the area is in need of extensive methodological advancement. The following checklist is intended to be a contribution to such development. Not every item in the check list introduces new issues for policy makers or computational epidemiologists, but, depending on the reader’s area of expertise, one or two are highly likely to be more significant than the others. Much of it is part of the folklore of the area, and could be classified as procedural and pragmatic know-how. More specifically, the contribution is to have these items made explicit as one concise list, and tied to working procedures as demonstrated by our workflow description (Figure 1).

#### 1. *All population data sets are regional*

To have access to data on the entire population on the planet is not a realistic goal. Hence, most studies are limited to one geographic region, such as a city, a state, or a country [24]. This means that the universe of discourse includes not only the individuals in this geographic region, but also that a certain proportion of the individuals must be allowed to leave the region. Moreover, visitors and immigrants from other regions should be included in the population data. Some computational epidemiology projects employing micro data use census data, others extrapolate from samples, and yet others use synthetic data. In the rare cases where registry data is available for a large population – as is the case for the Swedish population – hard methodological questions must still be answered regarding the generalizability of results: which parts of a scenario execution in Sweden are likely to be analogous to ones in Norway, Iceland, or the state of Oregon?

#### 2. *Population data are sensitive*

Even after extensive post-processing, any data set with real population data is subject to privacy and integrity concerns. In almost all countries, this means that running a simulator with the data set is subject to applying to an ethics board. If approved, data must be kept safe and experiments may be run in designated facilities only. This makes replication studies difficult, and it also

restricts alignment studies to less interesting data sets.

### *3. Verifying the simulator is a serious engineering challenge*

To formally verify that the simulator produces adequate results, is free from programming bugs, and can handle the computational complexity of modeling large outbreaks is, in general, not possible. The software is too large, as is the variation of possible input values and the spectrum of sensitivity analyses. Extensive testing – varying the hardware environment and the parameter values, including the random seeds for stochastic processes – yields evidence for adequacy, but no guarantees. This does not entail that the simulator is without use, or not to be trusted, but merely that its construction and maintenance is an engineering challenge.

### *4. Validating the simulator output is hard*

Pandemics have been few and far between. Modeling a future scenario on a real outbreak of the past has been done with some success in the area of epidemiology. The structural properties of current and future societies may vary greatly from those studied in the past, however. Air travel, hygiene, and working conditions are three out of many factors that affect the spread of communicable disease and that vary greatly in the historical perspective. The low probability of catastrophic events such as a pandemic makes it very hard to validate any simulation experiment against real-world events.

### *5. Assumptions and hypotheses should be stated and controlled by the policy maker*

Placing assumptions on top of assumptions will only create a gap between the policy maker and the computational epidemiologist. As illustrated by the example of selecting different initially infected individuals, the description of a single assumption can be interpreted in multiple ways, and the implementation of different interpretations can diverge significantly from the respective intention. The complexity of communicating all assumptions implied by the decisions of the policy maker arises from the tremendous difficulty in identifying implicit assumptions at every step of development. Because every addition to the model carries the risk of modifying the interaction of existing parameters, ensuring that all assumptions have been made by the policy maker becomes a formidable challenge.

### *6. Triggering outbreaks in the simulator is nontrivial*

To implement a simulator that always produces outbreaks is easy. Increasing the infectiousness of a disease (as done, e.g., [17]) or the number of initially infected, quickly yields a disease pattern affecting the entire giant component, i.e., every individual connected to other individuals through the social network or by geographical proximity (cf. [25]), forming the largest connected subgraph of the population graph (cf. [26]). If such settings are inconsistent with empirical data, or with assumptions and hypotheses declared, however, then the adequacy of the model should be questioned. There is evidence for the fact that the initial stages of a pandemic require a different kind of modeling than the later stages [27]. It would therefore be naïve to think that increasing the number of initially infected – in order to trigger outbreaks in a larger proportion of runs – would not affect the model of the entire pandemic.

### *7. Hybrid models need constant refinement*

A model in which the micro, meso, and macro properties are integrated has the potential to mirror reality in a relatively accurate way. Under the proviso that model adequacy yields better prediction, one could discard the simplest

models in favour of such hybrid models. The level of ambition, however, comes at the price of the model never being finished, and model-dependent artifacts becoming more difficult to identify. Since the world to be modeled is a moving target, and since macro data can often be replaced by micro data as it becomes available, there are always refinements to be made. The devil is in the details.

## Acknowledgements

The authors would like to thank the current leader of the MicroSim project at the Swedish Institute for Communicable Disease Control, Lisa Brouwers. The authors also thank Olof Görnerup, Eric-Oluf Svec, the editor, and the anonymous reviewers for their constructive comments.

## References

- [1] R. Ross, “An application of the theory of probabilities to the study of a priori pathometry. Part I,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 92, pp. 204–230, Feb. 1916.
- [2] W. O. Kermack and A. G. McKendrick, “A contribution to the mathematical theory of epidemics,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 115, pp. 700–721, Aug. 1927.
- [3] M. Boman and E. Holm, “Multi-agent systems, time geography, and microsimulations,” in *Systems Approaches and their Application* (M.-O. Olsson and G. Sjöstedt, eds.), ch. 4, pp. 95–118, Springer, 2004.
- [4] S. Eubank, H. Guclu, V. Kumar, M. Marathe, A. Srinivasan, and N. Toroczkai Zand Wang, “Modelling disease outbreaks in realistic urban social networks,” *Nature*, vol. 429, pp. 180–184, 2004.
- [5] N. M. Ferguson, D. A. T. Cummings, S. Cauchemez, C. Fraser, S. Riley, A. Meeyai, S. Iamsirithaworn, and D. S. Burke, “Strategies for containing an emerging influenza pandemic in Southeast Asia,” *Nature*, vol. 437, pp. 209–214, Sept. 2005.
- [6] I. M. Longini, A. Nizam, S. Xu, K. Ungchusak, W. Hanshaworakul, D. A. T. Cummings, and M. E. Halloran, “Containing pandemic influenza at the source,” *Science (New York, N.Y.)*, vol. 309, pp. 1083–1087, Aug. 2005.
- [7] D. Balcan, B. Goncalves, H. Hu, J. J. Ramasco, V. Colizza, and A. Vespignani, “Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model,” *Journal of Computational Science*, vol. 1, no. 3, pp. 132 – 145, 2010.
- [8] J. M. Epstein, “Agent-based computational models and generative social science,” *Complexity*, vol. 4, no. 5, pp. 41–60, 1999.

- [9] Swedish Institute for Communicable Disease Control, “Microsim Source Code.” <https://smisvn.smi.se/sim/>, October 2010.
- [10] L. Brouwers, M. Boman, M. Camitz, K. Mäkilä, and A. Tegnell, “Microsimulation of a smallpox outbreak using official register data,” *Eurosurveillance*, vol. 15, no. 35, 2010.
- [11] F. Coelho, O. Cruz, and C. Codeco, “Epigrass: A tool to study disease spread in complex networks,” *Source Code for Biology and Medicine*, vol. 3, no. 1, 2008.
- [12] Statistics Sweden. <http://www.scb.se>, October 2010.
- [13] K. Popper, “Philosophy of science: A personal report,” *British philosophy in mid-century*, pp. 155–191, 1957.
- [14] I. Lakatos, “Science and pseudoscience,” in *Philosophical Papers vol. 1*, pp. 1–7, Cambridge University Press, 1977.
- [15] R. Thom, *Structural stability and morphogenesis: An outline of a general theory of models*. Addison-Wesley, 1993.
- [16] R. Axtell, R. Axelrod, J. M. Epstein, and M. D. Cohen, “Aligning simulation models: A case study and results,” *Computational & Mathematical Organization Theory*, vol. 1, pp. 123–141, Feb. 1996.
- [17] B. Roche, J.-F. Guegan, and F. Bousquet, “Multi-agent systems in epidemiology: A first step for computational biology in the study of vector-borne disease transmission,” *BMC Bioinformatics*, vol. 9, no. 1, 2008.
- [18] H. Liljenström and U. Svedin, eds., *Micro, meso, macro: Addressing complex systems couplings*. World Scientific, 2005.
- [19] J. Espino, M. Wagner, C. Szczepaniak, F. Tsui, H. Su, R. Olszewski, Z. Liu, W. Chapman, X. Zeng, L. Ma, Z. Lu, and J. Dara, “Removing a barrier to computer-based outbreak and disease surveillance – The RODS Open Source Project,” *MMWR Morb Mortal Wkly Rep.*, vol. 53 Supplement, pp. 32–39, September 2004.
- [20] M. Crubezy, M. O’Connor, Z. Pincus, M. Musen, and D. Buckeridge, “Ontology-centered syndromic surveillance for bioterrorism,” *Intelligent Systems, IEEE*, vol. 20, no. 5, pp. 26–35, 2005.
- [21] D. Abramson, B. Bethwaite, C. Enticott, S. Garic, T. Peachey, A. Michailova, and S. Amirrazi, “Embedding optimization in computational science workflows,” *Journal of Computational Science*, vol. 1, no. 1, pp. 41 – 47, 2010.
- [22] B. Cakici, K. Hebing, G. M., P. Saretok, and A. Hulth, “CASE: A framework for computer supported outbreak detection,” *BMC Med Inform Decis Mak*, vol. 10, no. 14, 2010.

- [23] M. E. Halloran, N. M. Ferguson, S. Eubank, I. M. Longini, D. A. T. Cummings, B. Lewis, S. Xu, C. Fraser, A. Vullikanti, T. C. Germann, D. Wagener, R. Beckman, K. Kadau, C. Barrett, C. A. Macken, D. S. Burke, and P. Cooley, “Modeling targeted layered containment of an influenza pandemic in the United States,” *PNAS*, vol. 105, no. 12, pp. 4639–4644, 2008.
- [24] D. L. Chao, M. E. Halloran, V. J. Obenchain, and I. M. Longini, Jr, “FluTE, a publicly available stochastic influenza epidemic simulation model,” *PLoS Comput Biol*, vol. 6, p. e1000656, Jan. 2010.
- [25] M. Youssef, R. Kooij, and C. Scoglio, “Viral conductance: Quantifying the robustness of networks with respect to spread of epidemics,” *Journal of Computational Science*, vol. In Press, pp. –, 2011.
- [26] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Review*, vol. 45, pp. 167–256, 2003.
- [27] E. Bonabeau, L. Toubiana, and A. Flahault, “The geographical spread of influenza,” *Proceedings of the Royal Society B*, vol. 265, pp. 2421–2425, 1998.